

REMARKS

Prior to entry of this response, Claims 1-29 were pending. In the Non-Final Office Action mailed July 24, 2006 Claims 1-29 were rejected. Claims 1, 8, 16, 18, 21-22, 27, and 29 have been amended to clarify that which the Applicants claim as their invention and not to otherwise narrow the claims in any manner. No claims have been added, or canceled. No new matter has been added by way of this amendment. For at least the reasons discussed below, Applicants submit that the pending claims are patentable over the prior art of record.

Claim Rejections - 35 U.S.C. § 103

Claims 1-5 are rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 6,412,069 to Kavan in view of Alexey Kirichenko's publication "F-Secure Kernel Mode Cryptographic Driver...." Claims 6-7 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Kavan in view of Alexey, in further view of U.S. Publication No. 2004/0078568 to Pham et al. Claims 8-21 are rejected under 35 U.S.C. § 103(a) as being unpatentable over WO 01/80482 to Eun in view of Alexey. In addition, claim 22-29 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Eun in view of Pham. Applicants respectfully traverse these rejections.

Perhaps it would be helpful to provide a brief understanding of the invention, to illustrate why the claims are not rendered obvious by the cited references. Briefly, the embodiments are directed towards protecting a computer system's operating system (OS). The OS may include a kernel binary and an OS user level binary. The user level binary includes such components as hardware device drivers, hardware abstraction layers, windowing graphical interfaces, menus, user interfaces, and the like. When the OS user level binary is generated, selected integrity data is also generated. Such integrity data may include but is not limited to, a digital signature, a hash associated with the user level binary, and the like. The integrity data is also generated for the kernel. The kernel is modified to include the integrity data associated with the user level binary and the kernel, such that the integrity data and the OS user level binary are strongly associated with a particular operating system build. The kernel further includes a tamper detection component that is

configured to examine the OS binary against its associated integrity data. If tampering is detected, the tamper detection component may provide a tamper detection message indicating which OS binary may have been modified.

Claim 1 recites a method for protecting an operating system. The method includes determining integrity data for an operating system binary, wherein the integrity data enables detection of a modification to the operating system binary, and modifying a kernel with the integrity data, wherein the kernel is operable to employ the integrity data to detect the modification to the operating system binary. After a careful review of the cited references, Applicants respectfully submit that the cited references do not disclose or suggest the limitations of at least claim 1.

For example, Kavsan discloses cryptographic service software embodied that electronically communicates with a standard operating system of a personal computer. The cryptographic service software performs cryptographic service in the kernel space of the operating system. See Kavsan's Abstract, and Figure. Thus, unlike the claimed invention, Kavsan does not disclose or suggest services on the kernel space or other components of the operating system.

It is clear that Kavsan does not determine integrity data for an operating system binary. Kavsan's cryptographic service software performs cryptographic services at the kernel space (but not on the kernel space - or any component of the operating system, including an operating system binary). Kavsan's cryptographic service software also discloses that its algorithms may be used to encrypt signals at the driver level, but nowhere does Kavsan disclose or suggest determining integrity data associated with an operating system binary. See Kavsan, Col. 2, lines 10-24; Col. 2, lines 61-67; and Col. 3, lines 5-15 and 20-27. Thus, Kavsan does not disclose a method of protecting the operating system, let alone determining integrity data associated with an operating system binary.

Moreover, the Office Action even concedes that Kavsan does not explicitly disclose determining integrity data and detection of a modification to the operating system binary, but states

that Alexey does disclose such limitations. However, after a careful review of Alexey, Applicants respectfully submit that Alexey does not disclose or suggest such limitations.

Unlike the claimed invention, Alexey merely discloses providing of an F-Secure kernel mode cryptographic driver (module) that is implemented within the kernel and available only to the kernel mode system drivers. See Alexey, page 4, paragraph 4.

When an OS loader attempts to load the module of Alexey, the module runs an integrity test and a number of cryptographic functionality self-tests. Alexey, page 7, paragraph 3. Thus, while Alexey performs integrity self-tests, it does not appear to determine any integrity data associated with an operating system binary. Moreover, Alexey does not disclose or suggest modifying the kernel with the integrity data, and clearly, neither does Kavsan. Rather, Alexey merely describes running the tests and making a decision based on the results. Thus, the cited references either alone or in combination (which combination the Applicants deny) clearly cannot support a *prima facie* rejection of at least claim 1.

Eun is directed towards protecting a file system, unlike the present invention which is directed towards, protecting an operating system, as claimed by the Applicants. As shown, the file system being protected by Eun, resides within the kernel level. See Eun, figure 2. Thus, as disclosed Eun appears to be protecting the contents of the file system, e.g., a home page (See Eun page 1, lines 35-36), and not an operating system binary.

The Office Action acknowledges that Eun does not explicitly disclose modifying an operating system kernel with the integrity data; however, the Office Action again points to Alexey to disclose this. However, as stated above, Alexey appears not to modify the kernel with the integrity data. Thus, Eun in combination with Alexey (the combination of which is denied) does not render the claims obvious. Thus, Applicants respectfully submit that, because the cited references do not support a *prima facie* rejection, claims 8-21 should be allowed to issue.

As to claims 22-29, they are rejected over Eun in view of Pham. While the Office Action acknowledges Eun does not disclose performing a tamper detection action if the first integrity data indicates tampering of the operating system binary, the Office Action states that Pham discloses this. However, after a careful review of Pham, the Applicants respectfully submit that Pham does not teach or suggest such limitations.

Instead, Pham is directed towards securing persistent data by providing a security file system layer interposed between the kernel and the file system. The secure file system layer selectively constrains data transfer operations initiated through the operating system kernel. See Pham, paragraphs 12-13. Pham does not disclose or suggest performing actions with integrity data for an operating system binary as claimed by at least claims 22-29. Thus, Eun in combination with Pham (the combination of which the Applicants deny) does not render the claims obvious. Thus, Applicants respectfully submit that, because the cited references do not support a prima facie rejection, claims 22-29 should be allowed to issue.

In addition, because dependent Claims 2-7, 9-17, 19-21, 23-28 depend from Claims 1, 8, 18, 22, and 29 respectively, these claims should also be in condition for allowance for substantially similar reasons. Thus, Applicants' respectfully submit that because claims 1-29 are allowable for the reasons stated above, they should each be allowed to issue.

CONCLUSION

By the foregoing explanations, Applicants believe that this response has responded fully to all of the concerns expressed in the Office Action, and believe that it has placed each of the pending claims in condition for immediate allowance. Early favorable action in the form of a Notice of Allowance is urged. Should any further aspects of the application remain unresolved, the Examiner is invited to telephone Applicants' attorney at the number listed below.

Dated: October 24, 2006

Respectfully submitted,

By *Jamie L. Wiegand*
 Jamie L. Wiegand
 Registration No.: 52,361
 DARBY & DARBY P.C.
 P.O. Box 5257
 New York, New York 10150-5257
 (206) 262-8900
 (212) 527-7701 (Fax)
 Attorneys/Agents For Applicant